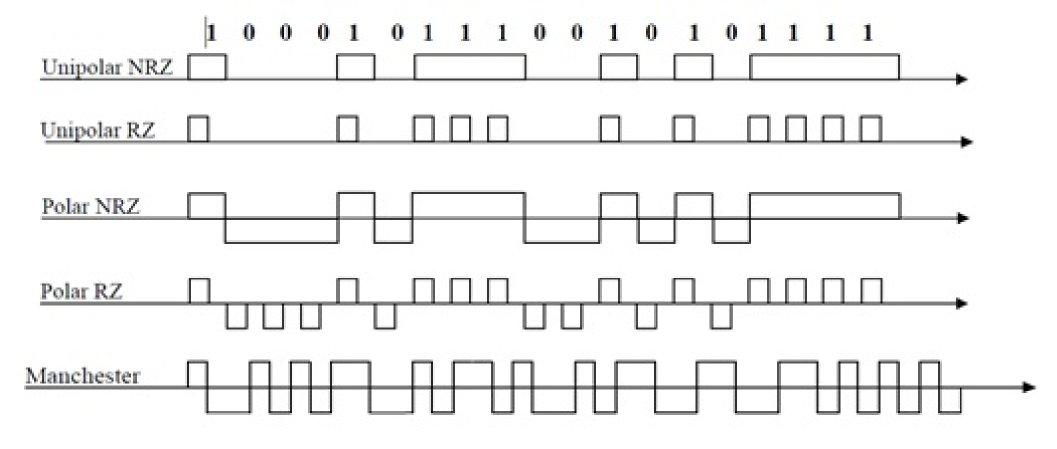
4.Códigos de Linea
En este módulo se presentan los códigos de línea usados comúnmente en un sistema de Comunicaciones Digitales: NRZ, RZ y Manchester. Se muestras las señales en tiempo y la Densidad Espectral de Potencia (DEP) de cada código. Se incluyen dos programas en GNU Octave: el primero introductorio que cuenta con una serie de preguntas iniciales, el segundo programa, permite al usuario seleccionar los parámetros de entrada de acuerdo a la prueba que desee realizar. Se presenta un video con la utilización del programa “Codigosdelinea” simulando un código de línea NRZp.

Códigos de Linea
Para poder transmitir por el canal de comunicación, una vez codificada la señal, es necesario convertir la secuencia de símbolos en una forma de onda adecuada para la transmisión bandabase denominada códigos de línea.

Aunque existen numerosos códigos de línea, los más conocidos son los llamados Retornos a Cero (RZ) y no retorno a cero (NRZ). El código NRZ mantiene constante el nivel de uno y cero durante todo el intervalo de bit. Si es polar, el uno y el cero tienen representaciones opuestas. En la codificación RZ, a la mitad del intervalo de bit el nivel de uno o del cero va a cero. También es conocido el código Manchester donde el símbolo uno se representa por medio de un pulso positivo seguido de uno negativo, ambos de igual amplitud y de medio símbolo de anchura; para símbolo cero las polaridades de estos pulsos se invierten.

En la figura 1 se muestran los códigos de línea más utilizados.





Códigos de Línea.

En la tabla 1 se muestran las expresiones de la Densidad Espectral de Potencia (DEP) de los distintos códigos de línea en el caso binario, asumiendo que los voltajes posibles son A y –A volts.

CODIGO	DEP G(f)
NRZ UNIPOLAR	$0.25A^2\delta(f) + 0.25A^2 t_b Sinc^2 ft_b$
RZ UNIPOLAR	$\frac{A^2}{16}[t_b Sinc^2 ft_b + \sum_n Sinc^2 \frac{n}{2}\delta(f + nf_b)]$
NRZ POLAR	$A^2 t_b Sinc^2 ft_b$
RZ POLAR	$0.25A^2 t_b Sinc^2(0.5ft_b)$
MANCHESTER	$A^2 t_b^2 Sinc^2(0.5ft_b)Sen^2(0.5\pi ft_b)$

Tabla 1. DEP Códigos de Línea

Si desea ampliar los contenidos de este módulo visite el sitio web desarrollado por la Prof. Trina Adrián de Pérez: Clase Códigos de Línea.

Descarga de Programas

Programas Códigos de Línea

Video

Codigos de Linea
http://www.youtube.com/v/bREalmz9ml0&rel=0

2. Codificación de Fuente
En este módulo se explicará detalladamente la teoría referente a codificación de fuente: modulación por código de pulso (PCM) y Modulación diferencial de Pulsos codificados(DPCM).Se incluyen dos programas en GNU Octave: el primero introductorio que cuenta con una serie de preguntas iniciales, el segundo programa, permite al usuario seleccionar los parámetros de entrada de acuerdo a la prueba que desee realizar. Se presenta un video con la utilización del programa “Codigosdefuente”.

Codificación de Fuente

Los sistemas de comunicaciones tienen como propósito transmitir información de fuente a destino. La fuente de información es una función variante en el tiempo que se modela mediante un proceso estocástico. En consecuencia, el sistema de comunicaciones está diseñado para transmitir la salida de un proceso estocástico (fuente de información) a un destino a través de un medio estocástico (canal). Las fuentes de información pueden clasificarse como fuentes analógicas o como fuentes discretas. Si la información, analógica o discreta, debe ser adaptada para trabajar con un sistema de comunicación digital, esta adaptación al sistema se conoce como Codificación de fuente, en este bloque se buscan técnicas para convertir de forma eficiente las fuentes arbitrarias de información en mensaje digital, eliminando la posible redundancia que presente la fuente de información.

Modulación de Pulsos codificados (Pulse- Code Modulation PCM)

Existen dos operaciones básicas en el proceso de conversión analógico digital de una señal: discretización en tiempo (muestreo) y discretización en amplitud (cuantificación).

Muestreo
El muestreo es un proceso lineal que permite transformar una señal de espectro limitado y continuo en el tiempo, en una serie (señal discreta) de valores de amplitud que constituyen sus muestras. Se puede representar

matemáticamente la señal muestreada como el producto de la señal analógica x(t) por un tren de impulsos, esto es lo que se llama muestreo ideal, ecuación 1.

Equation:

$$x_s(t) = \sum_{-\infty}^{\infty} x(nt_s)\delta(t - nt_s)$$

Donde x(t) es la señal muestreada, x(nts) es la amplitud de la muestra y ts es el período de muestreo o distancia entre muestras. En frecuencia, esta multiplicación en tiempo corresponde a una convolución entre el espectro de la señal analógica y el espectro del tren de delta, ecuación 2.

Equation:

$$X_s(f) = \frac{1}{t_s}\sum_{k=-\infty}^{\infty} X(f - kf_s)$$

Donde fs es la frecuencia de muestreo.

La ecuación 2 muestra que el espectro de la señal muestreada es la suma infinita de versiones escaladas y desplazadas múltiplos de fs de la señal original. Las versiones desplazadas no se solapan si la frecuencia de muestreo fs es mayor o igual a dos veces el ancho de banda de la señal analógica.

Cuantificación

La cuantificación es una operación no lineal, que consiste en dividir el rango total de la amplitud de la señal en M niveles de cuantificación de tamaño a, donde a es llamado el paso del cuantificador. Para cada instante de muestreo se determina en que intervalo de voltajes está la señal y, en base a esto, se le asigna uno de los M niveles de cuantificación.

La cuantificación puede ser uniforme o no uniforme cuando los pasos se eligen del mismo tamaño o no respectivamente.

Dependiendo de la aplicación y del receptor se eligen los niveles de cuantificación. Si se escoge M= 2n una señal cuantificada con M niveles puede ser convertida en una señal binaria donde cada nivel se representará con n bits; esto implica una velocidad de transmisión n veces mayor que la frecuencia de muestreo y por consiguiente un incremento del ancho de banda.

La cuantificación genera un error, llamado también ruido de cuantificación ε, definido como:

Equation:

$$\varepsilon = x_q(nt_s) - x(nt_s)$$

Donde xq es la señal cuantificada y x es la señal muestreada. La relación señal a ruido de cuantificación se define como:

Equation:

$$\left(\frac{S}{N}\right)_Q = \frac{E[x^2]}{E[\varepsilon^2]}$$

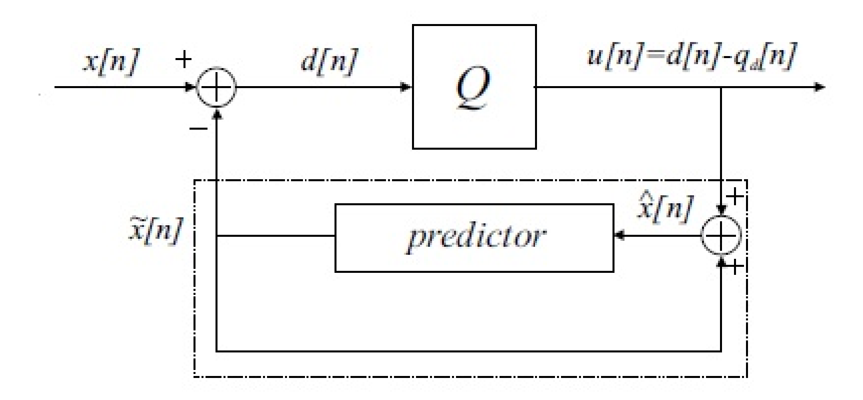
Donde E[x2] es la Potencia Promedio total de la Señal E[e2] es laPotencia Promedio del error o ruido de cuantificación

Si x(t) tiene una distribución probabilística no uniforme conviene usar un cuantificador no uniforme; es decir, uno que tenga pasos más estrechos en aquellas zonas de voltaje más frecuentes y pasos más grandes en zonas menos probables.

Aplicar una cuantificación no uniforme es equivalente a pasar la señal a través de un compresor y aplicar a la señal comprimida un cuantificador uniforme.

Si se desea reducir la redundancia de la señal muestreada, haciendo uso de su comportamiento estadístico, se puede emplear DPCM el cual se describe a continuación.

Modulación diferencial de Pulsos codificados (Differential Pulse Code Modulation DPCM)
Los sistemas DPCM se basan en los sistemas PCM, pero realizan la cuantificación de la diferencia de cada muestra de la señal (x[n]) con una estimación o predicción de la misma de la misma. En la figura 1 se muestra un diagrama del cuantificador DPCM. Cuando el predictor produce una buena estimación de la señal de entrada, la entrada del cuantificador Q, llamada error de predicción, requerirá un cuantificador de menos niveles y se logra un ahorro en la cantidad de bits para representar la señal.



Cuantificador DPCM

El predictor estima el valor de una muestra basándose en muestras anteriores pesadas, el valor de estimación debe ser lo más cercano a la muestra correspondiente para así minimizar la potencia de error de cuantificación. Los pesos o coeficientes del predictor se calculan a través de las ecuaciones resultantes de minimizar el error de predicción.

Descarga de Programas

Programas Codificación de Fuente

Video

Codificación de Fuente
http://www.youtube.com/v/OUL6rp6iqvA&rel=0

3. Procedimiento de Ortogonalización Gram Schmidt

A la salida del bloque de codificación de fuente (PCM-DPCM) ya se dispone de una secuencia de símbolos o bits a los cuales hay que asignarles formas de onda temporales a fin de lograr una adecuada transmisión a través del canal. Para visualizar el problema de trasmisión de señales de una manera gráfica, simplificar los cálculos matemáticos y para resolver el problema de detección, cada forma de onda se puede representar en función de un conjunto finito de bases ortonormales uj(t). Cada forma de onda tendría asociada una cierta combinación de coeficientes sij (ecuación 1)
Equation:

$$s_i(t) = \sum_{j=1}^{n} s_{ij} u_j(t)$$

Donde Si(t) es la forma de onda que se desea representar.

Esto es similar a la representación de vectores en función de bases ortogonales.

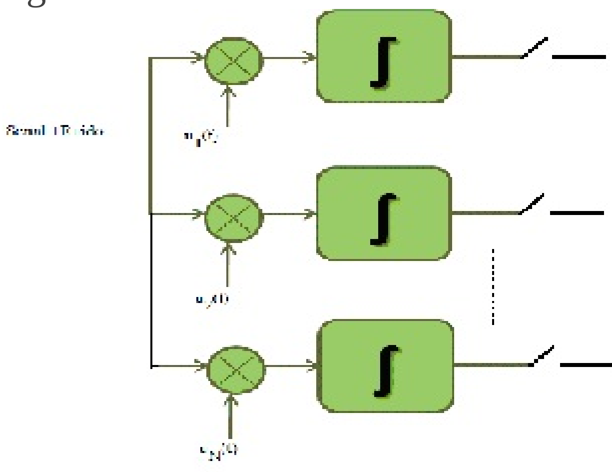
A través del procedimiento de ortogonalización Gram-Schmidt, las señales de energía pueden ser representadas por un conjunto de bases ortonormales que se derivan de la señal original.

Sea {s1,s2,...,sk} un conjunto de señales que representan una determinada transmisión, la familia de bases ortonormales {u1, u2,..., un}, ortogonales entre sí y de energía unitaria; se genera de acuerdo a la ecuación 2.
Equation:

$$u_n(t) = \frac{s_n(t) - \sum_{m=1}^{n-1} \langle s_n(t), u_m(t) \rangle u_m(t)}{\left\| s_n(t) - \sum_{m=1}^{n-1} \langle s_n(t), u_m(t) \rangle u_m(t) \right\|}$$

La representación de distintas formas de onda en un plano de N dimensiones definido por bases ortonormales se conoce como diagrama de

constelación. Usualmente en la práctica, para los tipos más usados de modulaciones, se implementan dos bases; en consecuencia los ejes que suelen ser llamados I (por In-phase y Q (por Quadrature). Los puntos en la constelación representan los símbolos de codificación o modulación que componen el alfabeto. La tasa de símbolos por segundo y la potencia de transmisión son parámetros que están inmersos en esta representación. Una de las ventajas de emplear el Procedimiento de Ortogonalización es el surgimiento natural de un receptor óptimo cuya estructura se muestra en la figura 1.



Receptor Óptimo

Una vez conocido el Procedimiento de Ortogonalización Gram-Schmidt se estudian las diferentes formas de señal a transmitir. Si son bandabase se habla de Códigos de Línea, si son Pasabanda de Modulación binaria o m-aria. Sin embargo, para mejorar la fortaleza frente al ruido se utiliza, antes de esos bloques, un sistema para compensar la redundancia perdida en el Codificador de Fuente llamado Codificación de Canal

Programas

Programas Ortogonalización Gram Schmidt

Si desea ampliar los contenidos de este módulo visite el módulo desarrollado por la Ing.Mezoa: Teoría básica de las Constelaciones.

Video

Ortogonalización Gram Schmidt
http://www.youtube.com/v/VgvyxzUHcm4&rel=0

7. Codificación de Canal
En este módulo se presentan los conceptos de Codificación de Canal. Se incluyen dos programas en GNU Octave: el primero introductorio que cuenta con una serie de preguntas iniciales, el segundo programa, permite al usuario seleccionar los parámetros de entrada de acuerdo a la prueba que desee realizar. Se presenta un video con la utilización del programa "codigosdebloques" .

Codificación de Canal

En los sistemas de comunicación digital, la señal puede verse afectada por diversas fuentes como el ruido introducido por el propio canal de comunicaciones y por el ruido de cuantificación, consecuencia del proceso de codificación.

La finalidad de la codificación de canal es la detección y corrección de errores producidos en el mismo, debido al ruido y distorsión introducidos por el medio de propagación y por las no linealidades en el propio sistema de transmisión.

Para medir la fidelidad de la transmisión de información se usa la probabilidad de error que es la probabilidad de que los símbolos a la salida del receptor sea diferentes a los símbolos transmitidos.

Seguidamente se revisan los tipos más usados de codificación de canal: codificación por bloques y codificación convolucional

Códigos de Bloque

Son códigos de longitud fija en el cual cada bloque está constituido por un número fijo de símbolos de información a los que se agrega una cantidad fija de símbolos de paridad. Mediante los bits de paridad se agrega redundancia a la información con el fin de detectar y corregir errores; en consecuencia, el ancho de banda aumenta reduciendo la eficiencia de transmisión. Entre los códigos de bloques más conocidos se encuentran los códigos Hamming.

Códigos Convolucionales

A pesar del buen rendimiento de los Códigos de bloques, desde el punto de vista de la Probabilidad de Error resultan pesados computacionalmente. La codificación convolucional es un proceso con memoria más sencillo, ya que opera de forma secuencial a nivel de cada bit tomando también en cuenta la información pasada. Los símbolos a la salida del codificador son interdependientes.

En los códigos convolucionales se definen tres parámetros: n es el número de bits a la salida del codificador, k el número de bits de información a la entrada de éste y m el número de registros de memoria. La relación k/n es la relación o tasa de código que proporciona una medida de la eficiencia de decodificación en cuanto al ancho de banda.

Una virtud adicional de la Codificación Convolucional es disponer de un decodificador óptimo, basado en el Diagrama Trellis, y denominado algoritmo de Viterbi.

Si desea ampliar los contenidos de este módulo visite el sitio web desarrollado por la Prof. Trina Adrián de Pérez: [Clase Codificación de Canal].

Programas

[Programas]

Video

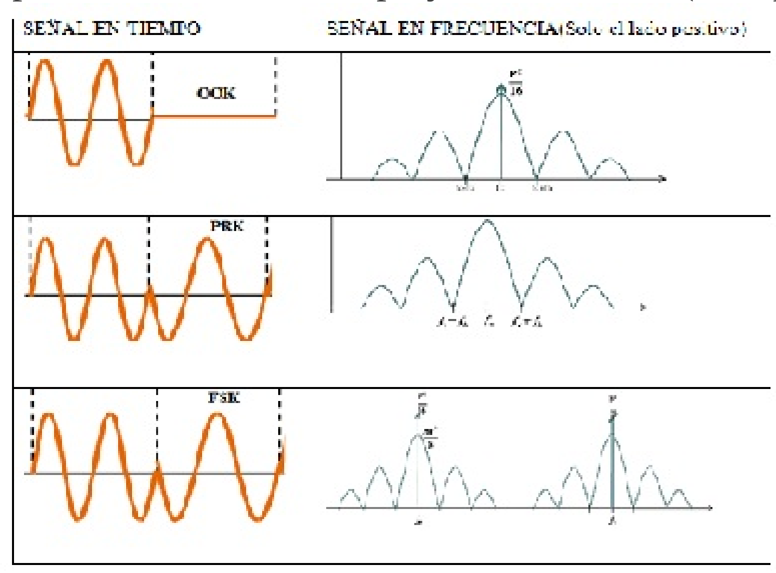
Codificación de Canal por Bloques
http://www.youtube.com/v/gfJ3voTroeI&rel=0

5.Modulación binaria y m-aria

En este módulo se presentan los conceptos de Modulaciones Binarias y M-arias. Se incluyen dos programas en GNU Octave: el primero introductorio que cuenta con una serie de preguntas iniciales, el segundo programa, permite al usuario seleccionar los parámetros de entrada de acuerdo a la prueba que desee realizar. Se presenta un video con la utilización del programa “modulaciones” .

Las técnicas de modulación modifican un parámetro de la señal portadora (Amplitud, frecuencia, fase) en función del mensaje con el fin de compartir el canal de transmisión, disponer de antenas con dimensiones razonables y mejorar la resistencia contra ruido e interferencia.

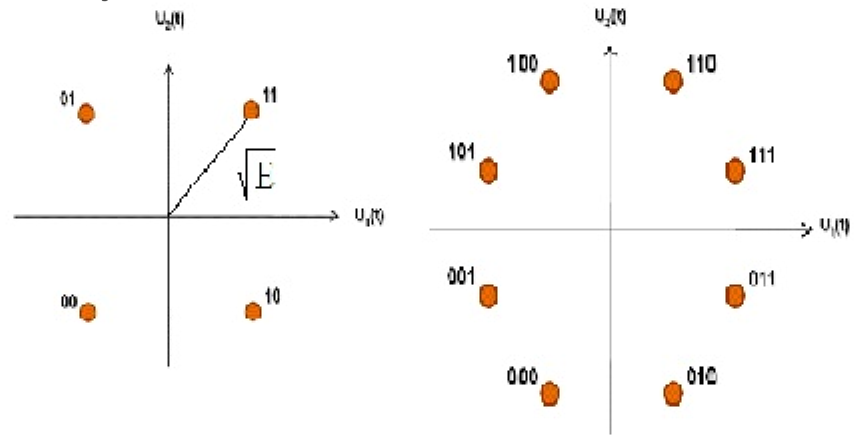
Las señales moduladas binarias pueden ser moduladas por amplitud frecuencia y fase (ASK, FSK, PSK). En la figura 1 se hace un resumen de casos particulares de ASK (OOK), PSK (PRK) y FSK. Se muestran sus representaciones en tiempo y en frecuencia (lado positivo)



Modulaciones Binarias

Las modulaciones M-arias (QAM, QPSK, MFSK) permiten entradas digitales con más de dos niveles de modulación. Una representación del

diagrama de constelación se muestra en la figura 2 para las modulaciones QPSK y 8PSK.



Si desea ampliar los contenidos de este módulo visite el sitio web desarrollado por la Prof. Trina Adrián de Pérez: Clase Modulaciones Binarias. y Clase Modulaciones M-arias.

Programas

Programas

Video

Modulaciones Binarias y M-arias
http://www.youtube.com/v/5rYlqxIZaRc&rel=0

1.Introducción a GNU Octave
En este módulo se presentan una introducción al software libre GNU Octave. Se incluyen dos videos para la instalación de Octave 3.6.1 y GUI Octave y Octave UPM.

GNU Octave

Es un lenguaje de programación de alto nivel, usado principalmente para la resolución de cálculos numéricos de alta complejidad. Provee capacidades para la resolución numérica de problemas lineales y no lineales, posee extensas capacidades gráficas, tanto para la visualización de datos como para su manipulación y permite la ejecución de scripts. Implementa una interfaz de comandos interactivos de línea, su lenguaje es compatible con C++, Fortran y su sintaxis es compatible con MATLAB. Es ampliamente utilizado por el entorno docente.

Es un software de distribución libre apegado a las condiciones de General Public License (GNU); es posible tener acceso al programa y al código fuente del programa para modificarlo. Funciona con sistemas operativos como Linux, Windowsy MacOS X.

Además de utilizar Octave para la enseñanza numérica, programación informática, simulación, modelización, también es usado en una amplia variedad de aplicaciones de investigación industrial y académica, incluyendo el procesamiento de señales, estadística genética, visión por computador, la econometría, la neurociencia y la biología computacional.

Octave es una aplicación orientada a la línea de comandos, no posee interfaz gráfica. Debido a esto se han desarrollado diversas opciones para instalar una interfaz gráfica compatible con el programa para que sea amigable al usuario y así sea posible generar un aprendizaje sencillo e intuitivo. En la Figura 1 se observa la línea de comandos para la versión de Octave 3.6.1.

```
Octave 3.6.1
GNU Octave, version 3.6.1
Copyright (C) 2012 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type `warranty'.

Octave was configured for "i686-pc-mingw32".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.

For information about changes from previous versions, type `news'.

warning: gmsh does not seem to be present some functionalities will be disabled
warning: dx does not seem to be present some functionalities will be disabled
warning: function C:\PROGRA~3\OCTAVE~1.1\share\octave\packages\statistics-1.1.0\
fstat.m shadows a core library function
octave:1>
octave:1>
```

Línea de Comandos Octave 3.6.1

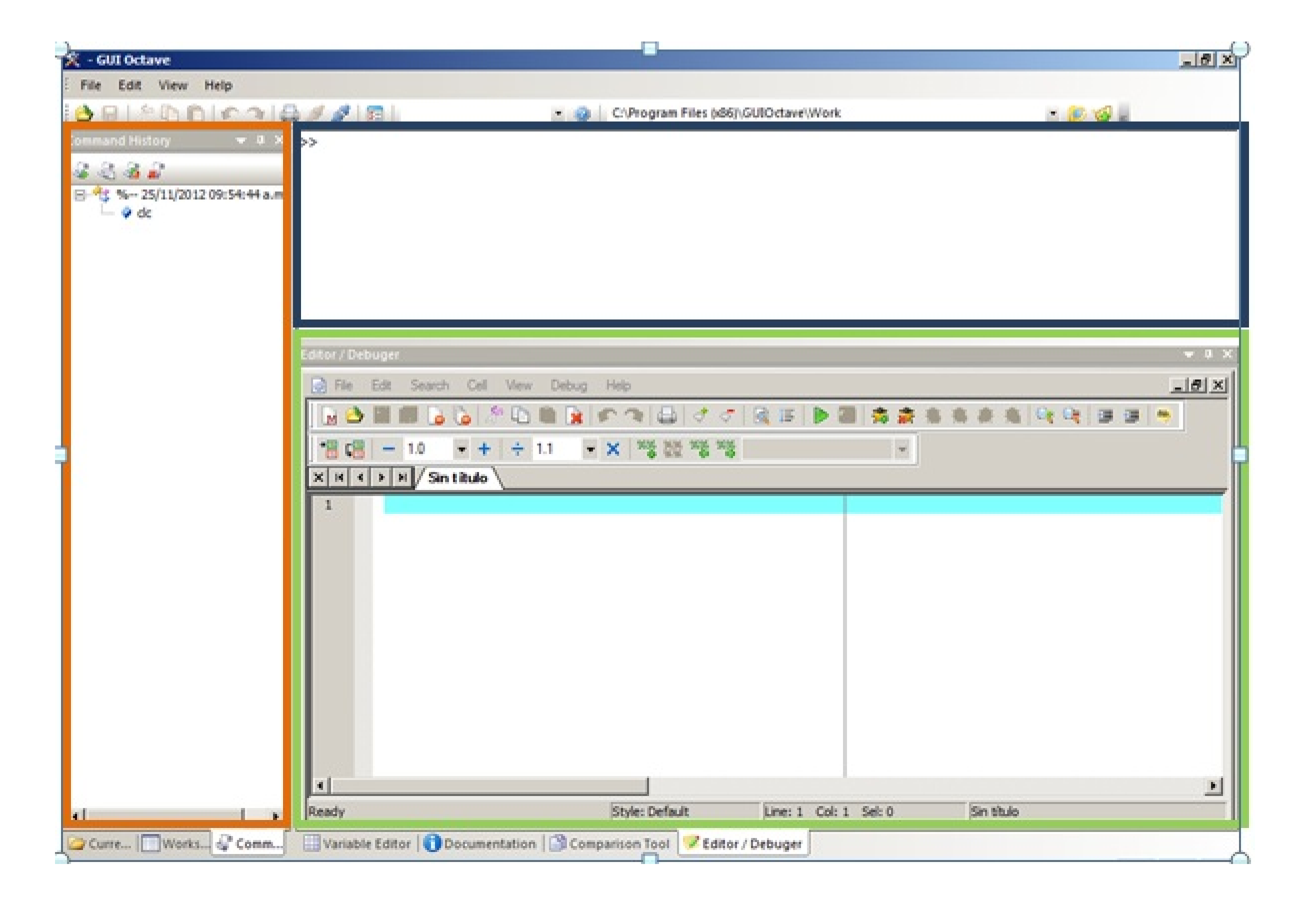
Las interfaces desarrolladas para Octave presentan características muy similares, brindan al usuario un conjunto de funcionalidades e interactividad, pero a su vez poseen limitaciones estructurales y deficiencias funcionales. Entre algunas de las interfaces desarrolladas se encuentran, GUI Octave, Koctave, Joctave, Goctave, YAOG, Emacs, Octivate, Octave Workshop y QtOctave.

Las interfaces están diseñadas para ofrecer un entorno gráfico al usuario de GNU Octave que cuenten con menús, botones y ventanas de diálogo. En la figura 2 se presentan las ventanas básicas que presentan las interfaces gráficas.

Ventana	Característica
Ventana De Control	Ejecución de los comandos
Editor	• Edición de textos • archivos.m • Permite crear modificar y ejecutar archivos.
Historial De Comandos	Visualización de comandos ejecutados en la ventana de control y en el editor
Help	Documentación de todos los comandos y funciones de Octave

Ventanas principales de interfaces gráficas de Octave

En la Figura 3 se observan las ventanas de la interfaz gráfica GUI Octave, compatible con la versión de Octave 3.6.1.En color naranja se identifica el historial de comandos, en color azul la ventana de control y en color verde el editor.



Ventanas de GUI Octave

Instalación

Si desea instalar la versión de Octave 3.6.1 visite la pagina "Octave for Windows": Octave 3.6.1.

Para descargar GUI Octave: GUI Octave.

Si desea instalar la versión de Octave desarrollada por al Universidad Politécnica de Madrid (octaveUPM): OctaveUPM.

Videos de Intalación

Intslación de GNU Octave 6.3.1
http://www.youtube.com/v/5ZRQ56Ol7Oo&rel=0

Intalación de GUI Octave
http://www.youtube.com/v/RiTjNMdSdR4&rel=0

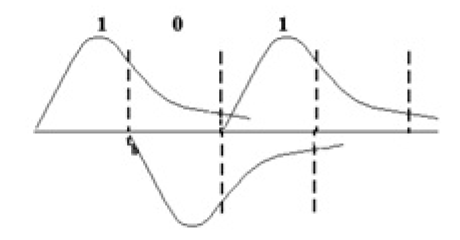
Una vez realizada la instalación de GNU Octave 3.6.1 y de el GUI Octave se debe reiniciar el Ordenador
Instalación de Octave UPM
http://www.youtube.com/v/ax08k5TCdfk&rel=0

Descargas

Carpeta Share : share

6.Interferencia Intersimbólica
En este módulo se presentan una introducción a los conceptos de ISI. Se incluyen un programa en GNU Octave donde se observa la capacidad que tiene un filtro coseno alzado para combatir ISI.

Para transmitir una señal digital, independiente del código de línea usado, se necesita un canal de ancho de banda infinito. Sin embargo, el canal tiene ancho de banda finito, por lo tanto los pulsos se esparcen y hará que estos entorpezcan la decisión sobre los bits vecinos. En la figura 1 se observan tres bits seguidos que se han dispersado debido a que el canal no tiene ancho de banda infinito. Podría ocurrir, que cualquiera de estos bits sea visto como un cero al llegar al receptor



3 Bits Dispersados

Criterios de Nyquist

La señal codificada y(t), se expresa como la convolución de una serie de impulsos aleatorios x(t) con un pulso determinístico p(t); para variar y(t) se puede variar x(t) o p(t). Laforma del pulso p(t) puede ser determinante para evitar la ISI.

Nyquist desarrollo criterios para seleccionar los pulsos de manera de controlar la ISI.

Primer Criterio de Nyquist:
Se desea un pulso recibido que tenga las siguientes características:

Equation:

$$p(t) = \begin{cases} 1 & t = 0 \\ 0 & t = \pm n t_b \end{cases}$$

$$\text{Ancho de Banda} = \frac{1}{2t_b}$$

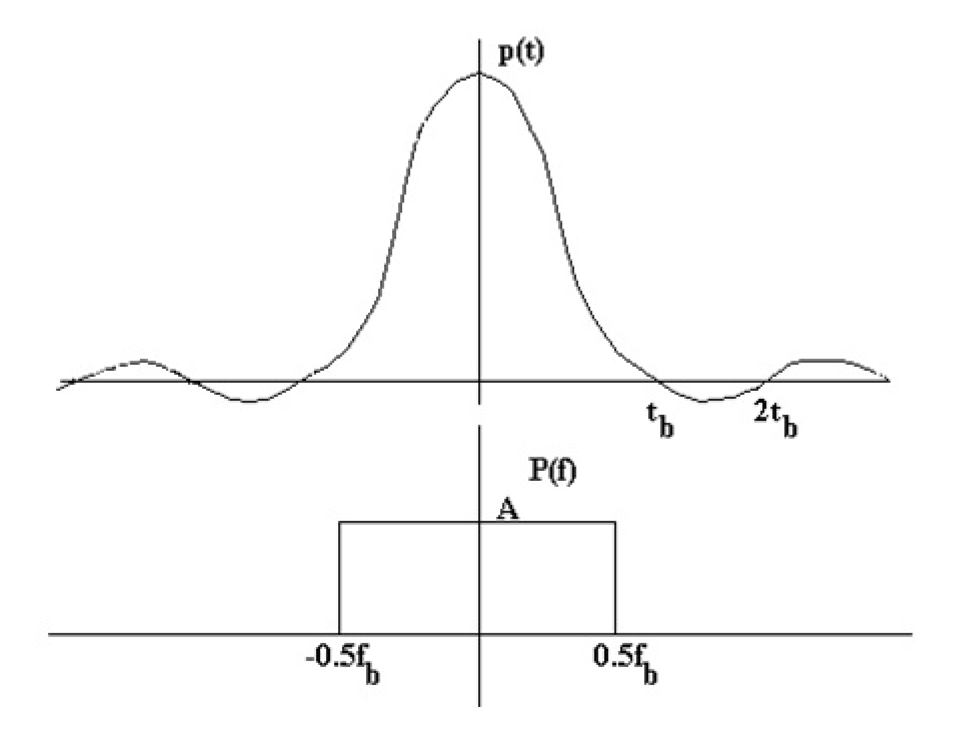
El pulso p(t) sería entonces

Equation:

$$p(t) = A f_b \,\mathrm{Sinc}\, t f_b$$

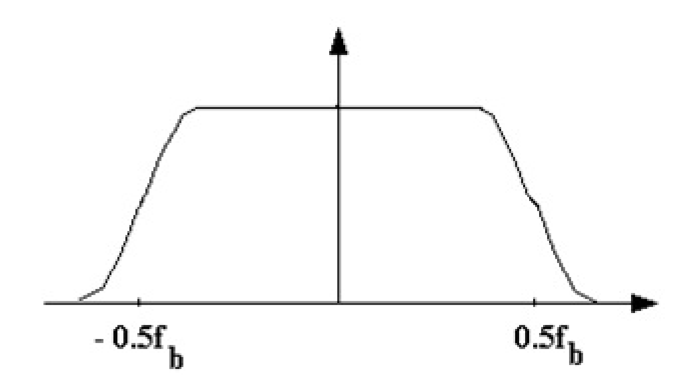
$$P(f) = A \prod (f t_b)$$

En la figura 2 se observa la forma del pulso en tiempo y su espectro en frecuencia



Pulso en tiempo y Frecuencia

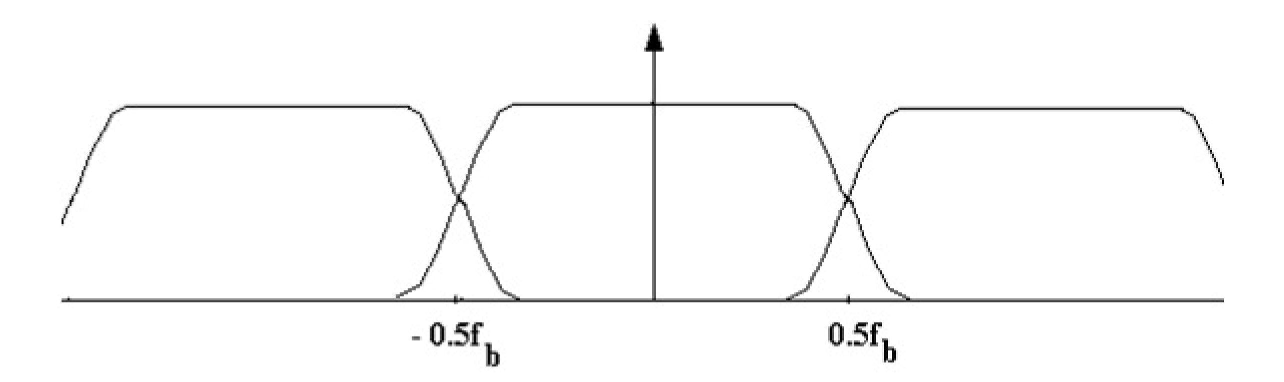
Este tipo de pulso no es realizable, flexibilizando el requisito de ancho de banda; por ejemplo se puede permitir un P(f) que ocupe un ancho de banda mayor que 0.5fb (figura 3)



En el receptor se muestrea cada tb. Se desea que al muestrear cada pulso y sus vecinos, solo quede el valor del pulso en el instante de muestreo de interés. Si se toma el valor en t=0 esto implica que al sumar todas las repeticiones de P(f) cada fb estas deben sumar una constante (figura 4)

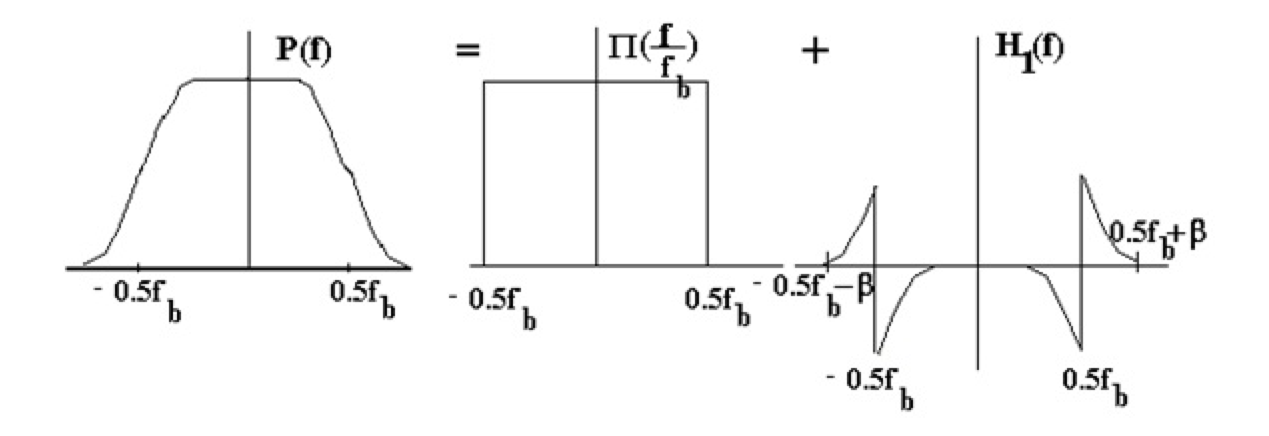
$$p(t)\sum \delta(t - kt_b) = p(0)\delta(t) = \delta(t)$$

$$P(f) * \frac{1}{t_b}\sum \delta(f - nf_b) = \frac{1}{t_b}\sum P(f - nf_b) = 1$$



A este tipo de filtros se les llama filtros vestigiales.

Para analizar matemáticamente el efecto, es conveniente descomponer P(f) en dos componentes :



3 Bits Dispersados

H1(f) tiene simetría alrededor de 0.5tb

$$P(f) = \prod\left(\frac{f}{f_b}\right) + H_1(f)$$

$$p(t) = \frac{1}{t_b} \mathrm{Sinc}\,\frac{t}{t_b} + h_1(t)$$

El Sinc se anula cada t=n tb excepto en t=0 que vale 1. Observe que H1 (f) es simétrica y par, por lo tanto:

$$h_1(t) = \int_{-\infty}^{\infty} H_1(f) e^{j\omega t} df = 2\int_{0}^{\infty} H_1(f) \cos\omega t \, df$$

$$h_1(t) = 2 \int_{0.5f_b-\beta}^{0.5f_b} H_1(f) \cos\omega t \, df + 2 \int_{0.5f_b}^{0.5f_b+\beta} H_1(f) \cos\omega t \, df$$

$$h_1(t) = -4 \mathrm{Sen}(2\pi 0.5 f_b t) \int_{0}^{\beta} H_1(0.5f_b + x) \mathrm{Sen}(2\pi x t) dx =$$

Observe que el término sinusoidal fuera de la integral se anula en cada ntb. Por lo tanto, como

$$p(t) = \frac{1}{t_b} \mathrm{Sinc} \frac{t}{t_b} + h_1(t)$$

el cual vale cero para todo t= ntb y toma el valor de 1/ tb para t=0 . Esto evita la interferencia inter simbólica.

Segundo Criterio de Nyquist:
Buscando eliminar la Interferencia inter simbólica y disminuir el ancho de banda, se define el segundo criterio de Nyquist, el cual se basa en definir los pulsos de manera que exista interferencia controlada entre un bit y sus vecinos mas cercanos. Conociendo la ley de interferencia uno puede detectar cada bit en el receptor. La señal ocupará menos ancho de banda pero consumirá mas potencia.

Programa

Programa

Video

Interferencia Intersimbólica (ISI)
http://www.youtube.com/v/fEiXrUDJ7I4&rel=0